(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/77897 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/26508

(22) International Filing Date:
26 September 2000 (26.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10-1999-0041359
27 September 1999 (27.09.1999) KR

(71) Applicant: **TORVEO TECHNOLOGIES** [US/US]; 201 Marshall Street, Redwood City, CA 94063 (US).

(72) Inventors: **HEO, Hong**; 523-10, Sangmunl-dong, Dobong-gu, Seoul (KR). **PARK, Sun, Eun**; 101 dong 608ho, Kolong APT, Bangy-dong, Songpa-gu, Seoul (KR). **LEE, Jin, Bae**; 502ho, 547-3 Dogok-dong, Kangnam-gu, Seoul (KR). **KANG, Hee, Kook**; 201ho, 463-43, Chunho2-dong, Kangdong-gu, Seoul (KR). **SEO, Seong, Cheol**; 211 dong 101ho, Donga APT, Keoye-dong, Songpa-gu, Seoul (KR).

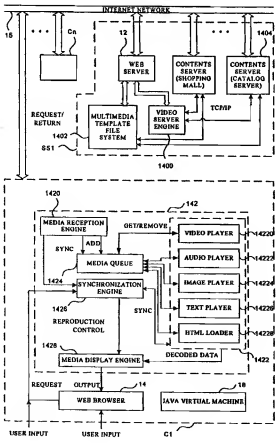
(74) Agents: **PIRIO, Maurice, J.** et al.; Perkins Coie LLP, P.O. Box 1247, Seattle, WA 98111-1247 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR WEB CATALOGING DYNAMIC MULTIMEDIA USING JAVA



(57) Abstract: A Java-based dynamic multimedia web cataloging system for expressing catalog contents most effectively with a web browser in which various multimedia data having streamed from a server system to a web browser is mixed with HTML documents. The system includes a web server for providing a client Java Applet information contained HTML catalog document to the web browser when user accesses the server system via the web browser, a video server engine for streaming the video data on a real-time basis, a template file system including template files having overall information appropriate for a streaming action, and the client Java Applet which is inserted in the HTML catalog document when the client terminal accesses the web server, downloaded in the web browser, and created by the Java virtual machine, for implementing functions necessary for a web cataloging dynamically on a real-time basis. The client Java Applet includes a synchronization engine, a media reception engine, a media player engine, and a display engine.

WO 01/77897 A2

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *without international search report and to be republished upon receipt of that report*

SYSTEM AND METHOD FOR WEB CATALOGING DYNAMIC MULTIMEDIA USING JAVA

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a system for web cataloging dynamic multimedia using Java (hereinafter referred to as Java) and a dynamic multimedia web cataloging method therefor, and more particularly, to a Java-based dynamic multimedia web cataloging system and method for expressing catalog contents most effectively with a web browser, in such a manner that various multimedia data having streamed from a server system to a client terminal using the web browser is mixed with hyper text markup language (HTML) documents, by inserting a Java Applet into a client application in which the Java Applet implements functions necessary for web cataloging multimedia dynamically on a real time basis in the static HTML documents.

Description of the Related Art

A set of contents expression methods for expressing static information including various added services and associated contents much more variously and compositely in a HTML document is called a multimedia web cataloging such as multimedia contents expression, public information, alignment, and advertisement. Infrastructures of Internet are expanded in various forms from an existing telephone line, to ISDN, ADSL, a cable modem, a satellite modem, etc., and thus the number of applications and utilities are explosively being increased including an electronic commerce (EC) web shopping mall or a tele-education.

The multimedia web cataloging can be efficiently used in an electronic commerce distributor and seller such as an Internet department store or shopping mall, for effectively displaying, advertising and selling products at low cost, or a cyber university for providing a tele-multimedia education.

Meanwhile, a dynamic multimedia web cataloging system means a system for composing, representing and managing multimedia contents and representing methods dynamically upon a user request on the Internet.

In general, a multimedia web cataloging system can be implemented in one of the following cases where: 1) only HTML documents are used, 2) Java Script or DHTML (Dynamic HTML) and 3) a plug-in or helper application is used.

Firstly, in the case where only the HTML documents are used to implement a multimedia web cataloging system, the simplest patterns of images or contents are output. Thus, in the cases where dynamic multimedia contents contained in a server are accessed in real time and displayed dynamically, or the composite contents are synchronized and output, the multimedia web cataloging system becomes inefficient or cannot be implemented with only the HTML documents. Therefore, a script program such as a separate Java script or a plug-in or helper application should be installed.

In the case that the Java script is used, a program function or size is limited often and a complicate multimedia representation and control is impossible.

In the case that the DHTML is used, it is appropriate for controlling and representing the simple pattern of multimedia but is limited often to represent a pattern of multimedia having a more intelligent procedure. In this case, it is too difficult for a general user to implement a

multimedia web cataloging system, and is also inefficient in view of the maintenance of the system.

In the case that a separate plug-in or helper application is used, all multimedia are integrated into a binary file and uploaded on a web server and the binary file is downloaded and executed using the plug-in or helper application. Accordingly, it is difficult to represent the binary file together with an existing information contained HTML document or the after-creation information. Further, whenever information of any one medium is changed, the overall file should be written or edited and then uploaded on a web server, which provides inconvenience in literary. Also, the user should install the plug-in or helper application program into his or her own web browser. Since the plug-in or helper application program is a unique application program, it has a drawback that the plug-in or helper application program cannot be combined in a desired position or pattern within a corresponding HTML document. That is, it is nearly impossible to change the layout of the plug-in or helper application in the HTML document.

Since most of the plug-in or helper application programs have been made to handle only successive media such as video or audio, it has many restricted points to combine them with discontinuous media such as an image, text or HTML document, to then be reproduced dynamically. Since the plug-in or helper program requires a different version program for every client specification, a complex investment or cost is needed for development and maintenance of the program.

SUMMARY OF THE INVENTION

To solve the above problems, it is an object of the present invention to provide a Java-based dynamic multimedia web cataloging system and method for expressing catalog contents most effectively with a web browser, in such a manner that various multimedia data necessary for current representation is requested dynamically to a server system according

to each feature of the multimedia data and having streamed via a web server from the server system to a client terminal using the web browser is mixed with hyper text markup language (HTML) documents, by inserting a Java Applet into a client application in which the Java Applet implements functions necessary for web cataloging multimedia dynamically on a real time basis in the static HTML documents.

It is another object of the present invention to provide a Java-based dynamic multimedia web cataloging system and method for effectively representing dynamic multimedia web cataloging information which was difficult to represent using a conventional HTML or DHTML (dynamic HTML) document, a plug-in program software and a Java script, in a web browser, in which a Java Applet inserted in a HTML document is driven by a client program and a synchronization engine and a streaming function for the Java Applet are properly mixed, to thereby logically transmit and receive and combine media frames such as video, audio, 2-dimensional/3-dimensional (2D/3D) image, text, animation and HTML data which have been selected on a real-time basis, through a web server and a video server engine.

It is still another object of the present invention to provide a Java-based dynamic multimedia web cataloging system and method which can be executed in a portable communications terminal such as PCSs and PHSs, a personal digital assistant (PDA), a settop box, a digital TV or a web phone which is porting a personal Java (PersonalJava) software, and a Java operating system (JavaOs) mounted Java chip, as well as a web browser of a personal computer (PC), a workstation, a notebook PC or a palm top PC which is designed based on a hardware-independent pure-Java and is mounted with a Java virtual machine.

It is a further object of the present invention to provide a Java-based dynamic multimedia web cataloging system and method which is configured on a streaming basis, in which only desired media can be selectively received according to an Internet bandwidth, and an intelligent

reproduction is possible using a frame-based virtual video cassette recorder (VCR).

It is still a further object of the present invention to provide a Java-based dynamic multimedia web cataloging system and method for providing a multimedia representation scheme which is the most appropriate on the web, in which a discontinuous media streaming method such as image, text or HTML data is newly devised from a conventional successive media streaming method such as videos or audios.

In the case that Java-based dynamic multimedia web cataloging system is used, a Java-based program can be automatically executed under nearly all the web browsers. Thus, if only one version Java-based program is uploaded in a web server, it can be driven in all web browser contained client's system without making a separate version for a different client. Also, a separate application program can be configured, a more intelligent procedure which cannot be implemented with a script language can be realized.

Also, in the dynamic multimedia web cataloging system, the shape or size of the Java-based program can be varied in the HTML document, and one or more replicas of an object is possible according to an object-oriented feature. Accordingly, a more variety and mixture of multimedia representation is possible in a single HTML document. Also, the Java-based program can be executed with a unique window as in a helper program and can facilitate a variety of multimedia representations which cannot be executed in a plug-in or helper program and mutual representation action with one or more various HTML documents, through an integral circumstance called Java.

To accomplish the above object of the present invention, there is provided a dynamic multimedia web cataloging system using Java under a client/server system circumstance where a client terminal including a web browser and a Java virtual machine and a server system including a web

server are connected via the Internet network, the Java-based dynamic multimedia web cataloging system comprising: a multimedia template file system configured with template files including overall information so that multimedia data including video, audio, 2D/3D image, animation, text and HTML data are appropriate for a streaming action; the web server for providing a client Java Applet information contained HTML catalog document to the web browser when a user gains access to the server system via the web browser, and for gaining accessing to the multimedia template file system directly for the multimedia data including audio, 2D/3D image, animation, text and HTML data in the case that there is a request from a client Java Applet created in the web browser according to the client Java Applet information and streaming the accessed result to the client Java Applet; a video server engine for parsing the contents of video data according to a request of the video data from the client terminal and streaming the video data on a real-time basis to the client Java Applet via the web server; and the client Java Applet which is inserted in the HTML catalog document when the client terminal gains access to the web server, downloaded in the web browser, and created by the Java virtual machine, for implementing functions necessary for a web cataloging dynamically on a real-time basis, wherein the client Java Applet comprises: a synchronization engine for requesting multimedia data including videos necessary for a corresponding logical point in time on the client web browser and combining the multimedia data received via the web server dynamically, to then reproduce the combined result; a media reception engine for receiving a cataloging multimedia data packet via a HTTP from the web server and the video server engine in a streaming method and temporarily storing the received cataloging multimedia data packet in a media queue; a media player engine for continuously checking virtual VCR information such as time information, reproduction information and velocity information of the synchronization engine and controlling corresponding media data to be read from the media

queue in a corresponding point in time and to be decoded in accordance with a format of each file to be reproduced in a corresponding arrangement device; and a display engine for displaying the media data decoded in the media player engine.

There is also provided a dynamic multimedia web cataloging method using Java under a client/server system circumstance where a client terminal including a web browser and a Java virtual machine and a server system including a web server are connected via the Internet network, the Java-based dynamic multimedia web cataloging method comprising the steps of: (a) gaining access to the web server of the server system using the web browser; (b) downloading a client Java Applet information contained HTML catalog document to the web browser from the web server; (c) parsing a HTML tag of the HTML catalog document in the web browser, requesting to the server system multimedia data including at least one of video, audio, 2D/3D image, animation, text and HTML data necessary for a multimedia web cataloging via a Java virtual machine, creating a client Java Applet object including a synchronization engine for combining the multimedia data received via the web server dynamically, to then reproduce the combined result and a media reception engine for receiving a cataloging multimedia data packet from the server system in a streaming method and storing the received cataloging multimedia data packet in a media queue, and creating a graphic user interface (GUI) in the determined HTML document; (d) parsing a user input parameter if a user initial input value is input via the web browser and then establishing flags with respect to media selected for a user desired catalog representation; (e) decoding each media data received in correspondence to each media selected based on the established representation information, in accordance with a format of each file, creating a media player object including a video player, audio player, an image player, a text player and a HTML player to be reproduced in a corresponding arrangement device, and allocating a media queue corresponding to each

media player object in a public type; (f) configuring initialization input data for starting a catalog representation in the client Java Applet in the case that the media player object has been created to be handed over to a Java event handler; (g) transferring a velocity value, a reproduction position and a frame rate in units of frame per second which are established according to a velocity control, and a selection of a reproduction position and a reproduction operation based on the parsed input value, from the synchronization engine to the server system via the web server; (h) parsing the contents in a video server engine upon a request of video data according to the request and streaming the parsed result to the client Java Applet via the web server and reading audio, image, text and HTML data except for video data from a template file system directly in the web server and streaming the read result to the client Java Applet; (i) receiving a cataloging multimedia data packet in the media reception engine in a streaming method and storing the received cataloging multimedia data packet in a media queue; (j) synchronizing each received media data under the control of a reproduction time of a synchronization engine in a media player, to be decoded in accordance with a format of each file to be reproduced in a corresponding arrangement device; and (k) displaying the media data decoded in the media player by a display engine.

As described above, the Java-based dynamic multimedia web cataloging system and method according to the present invention inserts a Java Applet into a client application in which the Java Applet implements functions necessary for web cataloging multimedia dynamically on a real time basis in the static HTML documents, for expressing catalog contents most effectively with the web browser, in such a manner that various multimedia data necessary for current representation is requested dynamically to a server system according to each feature of the multimedia data and having streamed via the web server from the server system to the web browser is mixed with the HTML document.

Also, the present invention can effectively represents dynamic multimedia web cataloging information which was difficult to represent using a conventional HTML or DHTML (dynamic HTML) document, a plug-in program software and a Java script, in a web browser, in which a Java Applet inserted in a HTML document is driven by a client program and a synchronization engine and a streaming function for the Java Applet are properly mixed, to thereby logically transmit and receive and combine media frames such as video, audio, 2-dimensional/3-dimensional (2D/3D) image, text, animation and HTML data which has been selected on a real-time basis, through a web server and a video server engine.

Further, the present invention can be executed in a portable communications terminal such as PCSs and PHSs, a personal digital assistant (PDA), a settop box, a digital TV or a web phone which is porting a personal Java (PersonalJava) software, and a Java operating system (JavaOs) mounted Java chip, as well as a web browser of a personal computer (PC), a workstation, a notebook PC or a palm top PC which is designed based on a hardware-independent pure-Java and is mounted with a Java virtual machine.

Still further, the present invention is configured on a streaming basis, in which only desired media can be selectively received according to an Internet bandwidth, and an intelligent reproduction is possible using a frame-based virtual video cassette recorder (VCR), and provides a multimedia representation scheme which is the most appropriate on the web, in which a discontinuous media streaming method such as image, text or HTML data is newly devised from a conventional successive media streaming method such as video or audio.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects and other advantages of the present invention will become more apparent by describing in detail the structures and operations

of the present invention with reference to the accompanying drawings, in which:

FIG. 1 is a block diagram schematically showing a general client/server system to which the present invention is applied;

FIG. 2 is a block diagram showing a client terminal which can be used in the present invention;

FIG. 3 is a block diagram showing a dynamic multimedia web cataloging system using Java according to a preferred embodiment of the present invention;

FIGs. 4A and 4B are flowchart view for explaining the operation of the synchronization engine included in the Java Applet of FIG. 3;

FIGs. 5A and 5B are flowchart view for explaining the operation of the media player included in the Java Applet of FIG. 3;

FIGs. 6A and 6B are flowchart view for explaining the operation of the server system of FIG. 3;

FIG. 7 shows a configuration of a multimedia template file system included in the server system of FIG. 3;

FIG. 8 shows a configuration of the video template file of FIG. 7;

FIGs. 9A and 9B are flowchart view showing the media reception engine included in the Java Applet of FIG. 3;

FIG. 10 shows a configuration of the media display engine included in the Java Applet of FIG. 3; and

FIG. 11 is a detailed block diagram of the multimedia hardware device included in the client terminal of FIG. 3.

DETAILED DESCRIPTION OF THE INVENTION

Preferred embodiments of the present invention will be described in detail with reference to the accompanying drawings.

Firstly, referring to FIG. 1 showing a general client/server system to which the present invention is applied, a plurality of client terminals C1-Cn each including a web browser 14 and a Java virtual machine (JVM) 18 and a plurality of server systems SS1-SSn each including a web server 12 are connected to each other via the Internet 15.

Any data processor including a multimedia processing function and a communications function enabling an Internet access by wire or wireless, may be used as each client terminal, for example, as shown in FIG. 2.

That is, a client terminal C1 includes a central processing unit (CPU) 2 and a main memory 3. The CPU 2 and the main memory 3 are connected to an auxiliary storage unit 4 such as a hard disc and a read only memory (ROM) 5 via a bus 10.

Program codes for assigning a command to the CPU 2 in corporation with an operating system (OS) are recorded in the auxiliary memory 4 and the ROM 5. The program codes are loaded in the main memory 3 and then executed.

In the client terminal C1, an input unit 7 such as a keyboard and a mouse for user interface, a display device 8 such as a cathode ray tube (CRT), a liquid crystal display (LCD) monitor and a beam projector, and a sound output device 9 such as a speaker are connected to the internal bus 10 via each controller (not shown).

Also, the client terminal C1 includes a communication modem, and a communication interface 6 such as a PCMCIA network card or an Ethernet network card, for communicating with a server system SS1. A web browser 14 and a JVM 18 are stored in the auxiliary storage unit 4, as programs necessary for connecting with the server system SS1 connected to the Internet network 15 via the communication interface 6. These programs are loaded in the main memory 3 and driven at the time of connecting with the Internet network.

As described above, any terminal having a communication function enabling the Internet access by wire or wireless and a multimedia processing function necessary for reproducing multimedia data received from the server system SS1 may be used as the client terminal C1.

Thus, a portable communications terminal such as PCSs and PHSs, a personal digital assistant (PDA), a settop box, a digital TV or a web phone which is porting a personal Java (PersonalJava) software, and a Java operating system (JavaOs) mounted Java chip, as well as a web browser mounted with a Java virtual machine (JVM), that is, a Java driven, web browser installed desktop computer are used as the client terminal C1.

Meanwhile, as shown in FIG. 3, the server system SS1 according to the present invention includes a web server 12 connected to the Internet network 15, a video server engine 1400 connected to the web server 12, for parsing the contents of video data according to a request with respect to the video data from the client terminal C1 and streaming the video data on a real time basis to a client Java Applet 142 via the web server 12, and a multimedia template file system 1402 connected to the web server 12, including template files designed to be adapted to stream multimedia data according to a request from the client terminal C1 to the client Java Applet 142 via the web server 12.

That is, the template files contain all information such as a store position, medium type, medium name, representation start time, and representation end time in the web server so as to be adapted to stream audio, 2D/3D image, animation, text, and multimedia data as well as video data.

Further, a media file directory of the multimedia template file system can share at least one contents server 1404 and a mutual physical store position, on the basis of a Transmission Control Protocol/Internet Protocol (TCP/IP) to be described later.

The server system SS1 provides a HTML catalog document containing Java Applet 142 information as shown in FIG. 3 to be described

later, via the web server 12 when a user gains access to the web server 12 using the web browser 14.

FIG. 7 shows a configuration of the multimedia template file system 1402 of the server system SS1. The multimedia template file system 1402 is divided into a template file directory 140202 and a media file directory 140204 physically.

The template file directory 140202 is a directory comprised of metadata handling actual data, which includes a video template file 140206, an image template file 140208, a text template file 140210, and a HTML template file 140212.

The video template file 140206 is comprised of a global information header (GlobalInfoHeader) 1402061, an MPEG header (MPEGHeader) 1402062, an MPEG data stream 1402063, an offset per frame (OFFSET_PER_FRAME) 1402064 and a time per frame (TIME_PER_FRAME) 1402065, in order to change the velocity of the input/output (I/O) of a conventional MPEG-1 video file, so as to be adapted to streaming.

Since the video template file 140206 is automatically loaded in the memory in the server system SS1 at the time when the server is initialized, streaming can start quickly when there arrives a request from the actual client terminal C1.

Other different media template files than the video template file 140206 have the following structure: a media file directory to take media information; a media type; a logical start time; and a logical end time.

Here, the media type is recognized using a file extension descriptor. For example, gif/jpeg represents an image type, txt represents a text, and html/htm represents a HTML document type. The template file can be written using a text-based editor by a user. In the present invention, since the contents of all the media differently from the conventional methods are

not integrated in a single binary file, correction of the changed items can be facilitated all the time.

The media file directory 140204 is a physical file system storing actual media data, which includes a video file 140214, an audio file 140216, an image file 140218, a text file 140220 and a HTML file 140222.

Also, as described above, the media file directory 140204 shares a mutual physical store position based on at least one contents server 1404 and the TCP/IP. Considering performance of the server system and load of the network, particular media data is designed so as to be dispersedly arranged in a particular contents server 1404.

Thus, a user can search a particular medium in a logical space. However, the particular medium can be located on a physical store position connected to one or more different networks. By doing so, the performance of the server system and the load of the network can be maintained low, and a limited storage space can be dispersed in a plurality of contents servers, to thus solve the drawback of a space limitation.

The dynamic multimedia web cataloging system of FIG. 3 according to the present invention starts to be driven when a user gains access to a web server 12 via a web browser 14 to download a HTML catalog document containing information of a client Java Applet 142 via the web browser 14. When the HTML tag of the document is parsed in the web browser 14 to find an Applet tag. Then, when the Applet tag is found, the web browser 14 produces an object of the client Java Applet 142 via an incorporated Java virtual machine (JVM) 18, and produces a graphical user interface (GUI) at a predetermined position in the HTML document.

The thus-produced client Java Applet 142 is implemented by the Java Applet capable of dynamically implementing functions necessary for a multimedia web cataloging such as multimedia contents expression, public information, alignment, and advertisement in a static HTML document on a real-time basis, inserted into the HTML document and driven together with a

server system SS1. The client Java Applet 142 includes a synchronization engine 1426 including a synchronization logic for requesting to the server system SS1 multimedia data of video, audio, 2D/3D image, animation, text and HTML data necessary for a corresponding logical point in time on the client web browser 14 having a Java virtual machine (JVM) 18, and for combining the multimedia data received via the web server 12 dynamically, to then reproduce the combined result, a media reception engine 1420 for receiving a cataloging multimedia data (such as video, audio, image, text and HTML data) packet from the web server 12 and a video server engine 1400 in a streaming method via the HTTP (HyperText Transfer Protocol) and storing the received cataloging multimedia data packet in a media queue 1424, a media player engine 1422 for decoding the received multimedia data according to each file format and reproducing the decoded result to a corresponding arrangement device and a display engine 1428 for reproducing the media data decoded in the media display engine 1422.

The roles of the synchronization engine 1426 will be described in more detail below. The synchronization engine 1426 performs a lifecycle management role for receiving information on a multimedia mixture from the Java Applet 142 inserted into the HTML document, synchronizing each medium selected and transmitted from the web server 12 on a logical point in time and managing each representation time, and a service manager role for parsing a template file containing all information such as a storage position, medium type, medium name, representation start time, and representation end time in the web server 142, relating to a web cataloging and dynamically expressing and managing objects of the parsed result in the Java Applet 142 and one or more HTML frames.

Here, each medium data transmitted from the web server 12 exists in units of frame. Each frame is comprised of attributes such as binary compression data, logical representation start time and logical representation end time to be actually represented.

The media player engine 1422 includes a video player 14220, an audio player 14222, an image player 14224, a text player 14226 and a HTML loader 14228.

Then, the client Java Applet 142 receives a user input via the web browser 14, generates a corresponding request from the synchronization engine 1426, and transfers the request to the server system SS1 via the web server 12. Here, the video server engine 1400 parses the contents of the request of the video data and streams the parsed result to the client Java Applet 142 via the web server 12, and the web server 12 reads the audio, image, text, HTML data directly from the multimedia template file system 1402 and streams the read result to the Java Applet 142.

The reason why a media stream group is divided into two kinds exists in the facts that a relative system load is minimized in operating a server system, in which video requiring a heavy system load is allocated by a dedicated processor to particularly use a server system streaming algorithm and media having relatively light loads other than video is streamed using a basic function of the web server 12.

Each multimedia data transmitted to the client Java Applet 142 as described above is temporarily stored in the media queue 1424 via the media reception engine 1420, and each of the media players 14220-14228 continuously checks virtual VCR information such as time information, reproduction information and velocity information of the synchronization engine 1426 and reads the corresponding media data from the media queue 1424 on the corresponding logical point in time, and decodes the read data via an internal decoder (not shown) and then outputs the decoded result to the web browser 14 via the corresponding media device of the media display engine 1428.

The respective media data which has been output once survives only for a logical reproduction time, and then after the logical reproduction time has been terminated, the media data is erased in the media queue 1424

by the corresponding media players 14220-14228. By doing so, a memory management can be efficiently performed.

Meanwhile, media to be reproduced within a respective corresponding logical time is classified into video, audio, text, 2D still image, 2D image set for 3D modeling, animation image and HTML data, which is independently performed at a mutually logical reproduction time in the window of the client Java Applet 142 and other web browser 14.

All the other media except for the 2D image set for 3D modeling are comprised of frames divided in units of an independent reproduction time. On the contrary, the 2D image set for 3D modeling is used in independent form for making a set of 3D modeling image and animation in the image player 14224.

In the case that video and caption are simultaneously reproduced as an exemplary and applicable example of the above media, a logical reproduction time is determined based on a frame rate per second (frames/sec) for each data frame of the video. In the cases of a caption text and a still image, a reproduction time is maintained only when the textual sentence or the contents are consistent with the video contents. On the contrary, in the case of audio data, a reproduction time of each data is determined according to a sample rate per second (samples/sec).

According to the above concepts, the multimedia template file system 1402 stores all information such as a storage position, medium type, medium name, representation start time, and representation end time in the web server 142 with respect to each media data as a text-based file so that a user easily inputs the information with a general text editor. The template file is used by the synchronization engine 1426 which refers to the template file to sets up a basic synchronization strategy when the client Java Applet 142 is firstly initialized.

Also, the other media data except for video data is stored in the multimedia template file system 1402 together with all the other information

which are physically separated from the media data, and are streamed independently by each of the media players 14220-14228.

If a 2D image set for 3D modeling whose data unit pattern is different from the other is used together, only the portion of the 2D image set is performed in the image player 14224, as described above.

When a web cataloging is performed dynamically in the above structure, the following media combinations are possible. Since the media combinations are run having unique synchronization information, more various and composite multimedia reproduction models can be presented. Each of the media can be independently performed and can be reproduced by linkage with one or more same media objects or different media objects.

Combination of media

-Video+textual caption

-Video+audio

-Video+audio+textual caption

-Video+audio+textual caption+still image

-Video+audio+textual caption (Korean)+textual caption (English)

-Video+audio+textual caption (Korean)+textual caption (English)+textual caption (Japanese)+...

-Video+audio+HTML document

-Video+audio+textual caption+HTML document

-Video+audio+textual caption+HTML document+still image

-Still image+textual caption□slide show

-Still image+textual caption+audio per each image□audio slide show

-Animation image□rolling banner advertisement

-Animation image+audio per each image□sound rolling banner advertisement

-3D panorama image

-Continuity of 2D images photographed at different angles□3D photo modeling effect

Particularly, in the case that a variety of HTML frames are used in the HTML synchronization, the HTML loader 14228 is implemented in the client Java Applet 142 so that a particular window name is made and retrieved.

As a result, a more variety of types of composite documents can be maintained, which provides a very efficient web cataloging.

The operational principles of the server system SS1 and the client Java Applet 142 will be described separately below in more detail.

First, the operation of the client Java Applet 142 will be described.

The client Java Applet 142 starts to be driven from a user initial input,

which can be received in the form of a tag or parameter inserted in the HTML data. Otherwise, the user input value can be received by retrieving public methods in the Java Applet 142 from a particular object in the HTML data via the Java Script.

Assuming that initial input values are received via a parameter value in the HTML data as a general case, the input values are chiefly GUI generation information, particular media selection, reproduction velocity establishment, initial random access, virtual VCR functions such as Play, Stop, Pause, FF and REW, and all information of media to be synchronized such as media type, server position, file name, total number of frames, the reproduction time of each frame or the position of the template file containing the information, in the client Java Applet 142. The initial input values are filtered via the synchronization engine 1426, by which a corresponding action is taken.

FIGs. 4A and 4B are flowchart view for explaining the operation of the synchronization engine 1426.

Referring to FIGs. 4A and 4B, when the client Java Applet 142 is initialized in the HTML data, a public type synchronization engine object is produced as an independent thread. The synchronization object thread is automatically performed with a demonstration process, and made to react whenever a user input value is generated (S1426).

Then, if a user initial input value is received (S142600), it is judged whether the input value has been input from the web browser 14 (S142602). If an input value has been input from the web browser 14, a user input parameter is parsed (S142606). Then, a user sets flags with respect to media selected for catalog representation (S142608) first of all.

If video has been selected, a flag "video_selected" is set to an "ON" value which is a full band variable of the client Java Applet 142. Likewise, the other media have the same process, respectively. If image media have been selected, it is additionally judged whether types of the selected image media are "JPEG," "GIF," or "Animated GIF." If no other items are input, all image types will be ready to be output basically.

In this way, if representation information for each medium has been set, media player objects are generated with respect to the media selected based on the set representation information (S142610). When each of the media players 14220-14228 is generated, a media queue 1424 is allocated as a public type, which configures a critical section using the synchronization function of Java.

After the selected media players have been generated, the client Java Applet 142 configures initialization input data for starting catalog representation (S142612) and hands over the configured result to a Java event handler. Also, dynamic user input values which are input during reproduction are input to the Java event handler via the client Java Applet 142 according to the judgement of the step (S142602) and parsed (S142604).

Then, the parsed input value is determined sequentially in the steps of velocity control (S142614), random access position value change

(S142616), play (PLAY) (S142618), pause (PAUSE) (S142620), stop (STOP) (S142622), fast forward/rewind (FF/REW) (S142624), show/hide (SHOW/HIDE) (S142626), and frame rate per second (FPS) (S14268). Then, a detailed step for a corresponding case is performed.

If an event relating to velocity control has been generated, an input value is set as a velocity value (S142638). A unit of the velocity value is configured with a decimal point first place based on 1.0. When a medium is reproduced in the reverse direction, a negative value is input. The input velocity value is transferred to the server system SS1 again, to thereby make the server system SS1 operate at the corresponding velocity (S142640).

In the case of moving to a desired random access reproduction position, the input value is set as a reproduction position value (S142642). A unit of the reproduction position value is configured in units of second. After an internal value with respect to the reproduction position has been changed, a display bar of the client Java Applet 142 is physically moved so that a user can track the changed value (S142644). In this case, the display bar can be omitted. At the same time, the changed position value is transferred to the server system (SS1), so that data can be read from the server system (S142646).

If an input value for PLAY is transferred, the velocity value is set as a normal velocity, that is, 1.0 (S142632). If an input value for STOP is transferred, the velocity value is set as 0.0 (S142634). If an input value for FF/REW is transferred, the velocity value is set as 3.0 or -3.0 (S142636). In the case of temporary stop (PAUSE), the velocity value is repeatedly set as 1.0 or 0.0 as a toggle function. Simultaneously, the changed position value is transferred to the server system (SS1), so that data can be read from the server system (S142646).

In addition, the user can show (S142648) or hide (S142650) a particular media player via the synchronization engine 1426. That is, only a desired medium can be dynamically represented any time during

reproduction. In this case, since the hidden (HIDE) media players do not receive data from the server system SS1 until the media players have been shown (SHOW), a network bandwidth can be efficiently used. Also, in the case that a frame rate per second FPS has been changed in the result of the judgement of step S142628, the changed FPS value is transferred to the vi□ server engine 1400 (S142652).

The synchronization engine 1426 having processed all the input items from the user opens its own object reference to each media player in order to manage and maintain synchronization of each player persistently. Through the opened reference, each media player can refer to synchronization variables and flags for current synchronization situation.

FIGs. 5A and 5B are flowchart view for explaining the operation of the media player. The media player 1422 can be configured with only media objects necessary for corresponding applications in order to reduce a program size and shorten a download time when the client Java Applet 142 is configured with jar files and installed in the web server 12.

For example, in the case that only a panorama function is used for configuring a catalog, only an image player 14224 object class is installed in the jar file in the FIGs. 5A and 5B steps. If an image and a text are used for configuring a slide show, only an image player class and a text player class are installed. If all functions are basically incorporated and a particular media class object is selected upon a user dynamic request, all classes are installed in the jar files.

The media players shown in FIG. 3 use a multithread function of Java basically. Thus, one or more selected media players can be performed in parallel and simultaneously. That is, referring to FIGs. 5A and 5B, a video player object generation step (S142210), an audio player object generation step (S142226), an image player object generation step (S142242), a text player object generation step (S142266), and a HTML player object generation step (S142284) are performed at the same time and

in parallel with each other. By doing so, one or more media can represented simultaneously.

The control flowchart of the media player will be described in detail with reference to FIGs. 5A and 5B. First, it is judged whether a user selects the output of video media (S142200). If the user has selected the video media, a video player object is generated as a public type (S142210). In this video player object generation step, the video player 14220 refers to the reference of the object of the already-opened synchronization engine 1426, to thereby obtain the latest synchronization information of the synchronization engine 1426 continuously. The object of the video player 14220 then generates a video decoder object for converting the latest synchronization information into pixel images which can be output in an actual media display engine 1428 (S142212). The video decoder object is generated into a thread object which can be independently performed.

The generated video decoder object checks if a video frame to be reproduced at a current point in time exists in the media queue 1424 (S142214). This judgement is accomplished by comparing the reproduction velocity and the reproduction position of the synchronization engine 1426, and the logical start time and the logical end time in the actual video frames. Here, if the velocity of the synchronization engine 1426 is zero, it is judged as a stop state and its thread is slept for α seconds in which $\alpha<1$ second (S142216). Then, the video decoder object goes to the step S142214 and repeats synchronization situation check continuously. If the velocity of the synchronization engine 1426 is not zero and there is no video frame corresponding to the current logical point in time, the video decoder object proceeds to step S142216.

If a video frame corresponding to the current logical point in time exists in the media queue 1424, the video frame is taken from the media queue 1424 (S142218), and the MPEG-1 decoding is performed (S142210). The video frames which can be reproduced in the current video player is

configured based on the MPEG-1 video stream. After the decoding has been completed, the decoded result is output on a screen by a video device of the display engine 1428 (S142222). Then, after a video decoder thread is slept for a remaining frame reproduction time so that the decoded output can be displayed on the screen for a reproduction time of the current frame, that is, a frame rate per second, the program proceeds to step S142214 in order to reproduce a next frame (S142224). In this case, the video frames which have been output once are automatically erased from the media queue 1424, to thereby reduce the load of the memory.

Meanwhile, it is judged whether audio media have been selected by the user in step S142202 just after judgment of step S142210. If the audio media have been selected, steps S142226-S142240 which are nearly similar to steps S142210-S142224 are performed. Only the difference resides in the fact that a basic type of the audio decoding supports the Au type of Sun Microsystems Co., Ltd. having a sampling rate of 8000Hz per second and the GSM (Global System for Mobile telecommunication) type in which the audio signals are compressed and transmitted into 13000bps. Since the GSM audio type where the audio signals are transmitted at 13kbps is supported, an audio support is possible on a real-time basis even with a 28K modem.

Meanwhile, it is judged whether image media are selected in step S142204. If the user has selected the audio media, an image player 14222 object is generated and the object reference of the synchronization engine 1426 is referred to (S142242). Here, an image decoder object is generated together with sub-decoding component objects which can selectively handle a 2D image modeling and a 3D image modeling according to a media application type (S142244 and S142246).

Although the video player 14220 and the audio player 14222 handle continuous media patterns, the image player 14224, the text player 14226 and the HTML loader 14228 handle discontinuous media patterns.

Accordingly, a corresponding template file of the multimedia template file system 1402 which has been described referring to FIG. 3 are read from the server system SS1 and the read template file is used in steps S142250, S142274 and S142290.

The multimedia template files 140208, 140210 and 140212 of the remaining discontinuous media except for the video are configured in the following pattern as described above. That is, the pattern includes a media file directory from which media are taken, a media type, a logical start time and a logical end time.

Here, the media type is recognized as a file extension descriptor. For example, gif/jpeg represents an image type, txt represents a text type, and html/htm represents a HTML document type. The template file can be made up with a text-based editor by the user. Since the contents of all media are not integrated in a binary file unlike the existing method, the altered items can be corrected any time.

Based on the above information, the actual media data are read from the server system SS1 in real time and combines the read result with a logical reproduction time to thereby complete a perfect media frame, through steps S142252, S142274 and S142294. The thus-completed media frame pieces are decoded through corresponding decoding steps S142260 and S142258, and represented on the display engine 1428 (S142262, S142280 and S142296).

After the media frames have been displayed on the screen, each decoder object thread is slept in order to ensure the reproduction time for each reproduction time, the media frames are erased from the screen if the reproduction time is finished unlike the continuous media to thereby ensure a discontinuity of the media (S142264, S142282 and S142298). In addition, the bandwidth of the network can be effectively used through the insurance of the discontinuity.

Here, a portion which should be taken care of is a 3D image modeling judgement step S142258 in the image decoder. This portion is divided into a motion module and a panorama module largely. The motion module receives the successive 2D images of 18 sheets or more which have been photographed at a different angle, in sequence and performs a three dimensional modeling, to thereby provide an effect of watching a 3D object. Further, a panning function is provided to sequentially change images, by which an actual three dimensional object seems to be seen.

The above function provides an effect of modeling a displayed product as its three dimensional shape in the web cataloging system. Also, the panorama module implements a three dimensional 360-degree panning effect with a sheet of 2D image. Also, when a particular space in the image is clicked, the panorama module includes a function of linking the image with a different media player.

FIGs. 6A and 6B are flowchart view for explaining the operation of the server system. The server system SS1 starts from step S140000 which drives video server servlets via the web server 12. Then, a video template file 140206 is read from the multimedia template file system 1402 and the read video template file is loaded in the memory. Then, each video template file is converted into an object, which can react to a user request with respect to a corresponding media player (S140002). The reason why the template file 140206 has been loaded in the memory resides in the fact that a user reaction time is reduced at maximum and a service performance is improved. When the template file is loaded, the actual contents data exists in the hard disc and only header portions containing the information of a physical storage position and the time information per each frame are installed in the memory.

In the case of the remaining media except for the video, that is, audio, image, text and HTML data, a corresponding file is read from the disc in step S140040 when a user request exists (S140038), the data is transmitted

to the client terminal C1 according to a streaming protocol with the client terminal C1 (S140042), to accordingly disconnect a session (S140044).

In the case of the video data, the video data is specially taken care of by the video server engine 1400, since the video data consumes a relatively large amount of data and a relative large number of input/output (I/O) resources, and reflects the situation of the network bandwidth which is dynamically varying and streams. Once a user performs a session connection with respect to a video servlet (S140004), the following commands is transmitted from the client terminal C1 to the video servlet in the form of the URL (uniform resource locator).

http://videoserver:port/VideoServlet?file=title1.mpgc&fps+10&lts=0&speed=1

The video server engine 1400 parses a video title name in the input commands (S140006), and selects a corresponding template object from the memory with respect to the parsed title (S140008). Then, a value of the reproduction position of the actual video data to be read is determined from the input parameter (S140010). If the reproduction position value is zero, the reproduction position is moved from the video template object to an initial offset pointer (S140012), and if not the reproduction position is moved from the video template object to an offset pointer of a corresponding position (S140014).

Then, it is judged whether a reproduction velocity is altered (S140016). If the reproduction velocity has been altered, the altered reproduction velocity is reflected (S140020), and if not the existing velocity is maintained (S140018).

Then, it is judged whether a frame rate (FPS: frames per second) is altered (S140022). If a current FPS input value on which the user reflects the situation in the alteration of the network bandwidth differs from

the existing value, the FPS value is set the newly input value (S140024), and if not the existing FPS is maintained (S140026).

If a parsing process with respect to all the input parameters are completed as described above, a video stream file is read from an actual current offset at a current velocity (S140028). The reading of the video data is repeated in units of one second. As soon as a one-second amount of the video data has been read, only a stream of the current FPS is packetized (S140030) and the packet is transmitted to the client terminal C1 (S140032). The above process is repeated until an EOF (end of file) is met (S140034) or a session is compulsively terminated (S140036).

FIGs. 9A and 9B are flowchart view for explaining the operation of the media reception engine. The media reception engine 1420 is implemented by producing a media reception engine object (S142000) when the client Java Applet 142 is driven. Once the object of the media reception engine 1420 has been produced, a packet of predetermined set α seconds is requested to the server system SS1 in advance in order to calculate the bandwidth of the current network. Here, the initial parameters such as a video title name, a FPS, a reproduction velocity=1, and a reproduction position=0 are transferred to the server system SS1 (S142002). Then, the media reception engine 1420 receives the packet of the α seconds from the server system SS1 and then calculates an average reception time value L per second (S142004).

Thereafter, an optimal FPS is calculated at the current network state with the average reception time value L (S142006). Accordingly, with this portion, the present invention system provides a mechanism which can selectively receive a conventional 30 FPS-based MPEG stream according to the current bandwidth situation. As a result, the present invention can provide elastical services with respect to the low bandwidth situation and the dynamic bandwidth situation which cannot be done in the conventional normal MPEG stream. Once the optimal FPS is calculated at the current

state, the system enters an asynchronous listen state until data is transmitted from the actual server system SS1. Thus, since the present invention system is in the wakeup state only when data is transmitted from the actual server system SS1, the opportunities that other processes can proceed are provided (S142008).

If a data packet is arrived from the server system SS1 (S142010), the data packet is received every second (S142012). Then, a media type is parsed at the header portion of the received packet (S142014). Then, the data packet is inserted into the corresponding media queue 1424 (S142016).

By repeating the above process, the successive packets are inserted into the media queue 1424. However, if a bandwidth situation becomes worse suddenly during reception, the logical end time of the received packet may exceed the logical time, that is, the reproduction position under the current reproduction (S142018). In this case, a step of buffering the time as much as a certain amount of time is needed. The buffering step starts when the reproduction velocity of the synchronization engine 1426 is set zero and enters a block state for a moment (S142020).

Then, an optimal buffering time is predicted based on the average reception value per second L (S142022). Then, a data packet is received as much as the buffering time and the received data packet is inserted into the media queue 1424 (S142024). If the buffering step is completed, the synchronization reproduction velocity is set one again and thus a reproduction is resumed from the stop point (S142026). The average reception time per second L is recalculated every five seconds (S142028). The media reception engine 1420 awaits the next packet at the listen state whenever the reception step of one packet is finished (S142008).

FIG. 10 shows a configuration of the media display engine. The media display engine 1428 is composed of three API, including a Java AWT (abstract windows toolkit) API (application program interface) 42802

taking charge of GUI, a Java audio API 42806 taking charge of audio output and a web browser API 42810 taking charge of the output of the HTML. The Java AWT API 42802 is connected to a video display device 42804 and takes charge of visual representation. The Java audio API 42806 is connected to an audio output device 42808 and takes charge of audio output. also, the web browser API 42810 performs the processes relating to the HTML output.

FIG. 11 is detailed block diagram of the multimedia hardware devices 8 and 9 shown in FIG. 2. The multimedia hardware devices 8 and 9 include a sound card 1602 and a VGA output unit 1606, respectively. The sound card 1602 is connected to a speaker system 1604 and converts the actual audio sound into an analog waveform to then output the results. The VGA output unit 1606 outputs the screen signals to a monitor 1608, a LCD 1610 or a beam projector 1612 according to a user selection.

As described above, the present invention provides a Java-based dynamic multimedia web cataloging system and method for expressing catalog contents most effectively with a web browser, in such a manner that various multimedia data necessary for current representation is requested dynamically to a server system according to each feature of the multimedia data and having streamed via a web server from the server system to a client terminal using the web browser is mixed with hyper text markup language (HTML) documents, by inserting a Java Applet into a client application in which the Java Applet implements functions necessary for web cataloging multimedia dynamically on a real time basis in the static HTML documents.

Also, the present invention provides a Java-based dynamic multimedia web cataloging system and method for effectively representing dynamic multimedia web cataloging information which was difficult to represent using a conventional HTML or DHTML (dynamic HTML) document, a plug-in program software and a Java script, in a web browser, in which a Java Applet inserted in a HTML document is driven by a client

program and a synchronization engine and a streaming function for the Java Applet are properly mixed, to thereby logically transmit and receive and combine media frames such as video, audio, 2-dimensional/3-dimensional (2D/3D) image, text, animation and HTML data which have been selected on a real-time basis, through a web server and a video server engine.

Further, the present invention provides a Java-based dynamic multimedia web cataloging system and method which can be executed in a portable communications terminal such as PCSs and PHSs, a personal digital assistant (PDA), a settop box, a digital TV or a web phone which is porting a personal Java (PersonalJava) software, and a Java operating system (JavaOs) mounted Java chip, as well as a web browser of a personal computer (PC), a workstation, a notebook PC or a palm top PC which is designed based on a hardware-independent pure-Java and is mounted with a Java virtual machine.

Also, the present invention provides a Java-based dynamic multimedia web cataloging system and method which is configured on a streaming basis, in which only desired media can be selectively received according to an Internet bandwidth, and an intelligent reproduction is possible using a frame-based virtual video cassette recorder (VCR) and provides a Java-based dynamic multimedia web cataloging system and method for providing a multimedia representation scheme which is the most appropriate on the web, in which a discontinuous media streaming method such as image, text or HTML data is newly devised from a conventional successive media streaming method such as videos or audios.

As described above, particular embodiments of the present invention have been described as preferable examples. However, the present invention is not limited to the above-described embodiments, and various modifications and changes can be possible by those skilled in the art without departing off the scope of the invention.

CLAIMS

What is claimed is:

1. A dynamic multimedia web cataloging system using Java under a client/server system circumstance where a client terminal including a web browser and a Java virtual machine and a server system including a web server are connected via the Internet network, the Java-based dynamic multimedia web cataloging system comprising:

a multimedia template file system configured with template files including overall information so that multimedia data including video, audio, 2D/3D images, animation, text and HTML data are appropriate for a streaming action;

the web server for providing a client Java Applet information contained HTML catalog document to the web browser when a user gains access to the server system via the web browser, and for gaining accessing to the multimedia template file system directly for the multimedia data including audio, 2D/3D image, animation, text and HTML data in the case that there is a request from a client Java Applet created in the web browser according to the client Java Applet information and streaming the accessed result to the client Java Applet;

a video server engine for parsing the contents of video data according to a request of the video data from the client terminal and streaming the video data on a real-time basis to the client Java Applet via the web server; and

the client Java Applet which is inserted in the HTML catalog document when the client terminal gains access to the web server, downloaded in the web browser, and created by the Java virtual machine, for

implementing functions necessary for a web cataloging dynamically on a real-time basis,

wherein the client Java Applet comprises:

a synchronization engine for requesting multimedia data including videos necessary for a corresponding logical point in time on the client web browser and combining the multimedia data received via the web server dynamically, to then reproduce the combined result;

a media reception engine for receiving a cataloging multimedia data packet from the web server and the video server engine in a streaming method and temporarily storing the received cataloging multimedia data packet in a media queue;

a media player engine for continuously checking virtual VCR information such as time information, reproduction information and velocity information of the synchronization engine and controlling corresponding media data to be read from the media queue in a corresponding point in time and to be decoded in accordance with a format of each file to be reproduced in a corresponding arrangement device; and

a display engine for displaying the media data decoded in the media player engine.

2. The dynamic multimedia web cataloging system using Java of claim 1, wherein said synchronization engine comprises:

lifecycle management means for receiving information on mixture of the multimedia data from the Java Applet inserted into the HTML document, synchronizing each of the selected media supplied from the web server on a logical time, and managing each representation time; and

service manager means for parsing the template file containing all the information relating to the web cataloging on a real time basis, and dynamically representing and managing the objects in the Java Applet and one or more HTML frames,

3. The dynamic multimedia web cataloging system using Java of claim 1, wherein each media data transmitted from said web server exists in units of frame, each frame including binary compression data to be actually represented, a logical representation start time and a logical representation end time, and wherein said all information includes a storage position, a media type, a media name, a representation start time and a representation end time in the web server.

4. The dynamic multimedia web cataloging system using Java of claim 1, wherein said multimedia template file system is comprised of a template file directory including metadata for handling actual media data and a media file directory which is a physical file system storing the actual media data,

said media file directory is comprised of a video file, an audio file, an image file, a text file and a HTML file,

said template file directory is comprised of a video template file, an image template file, a text template file and a HTML template file, which are automatically loaded into the memory in the server system and physically positioned in the header portion of the video file when the server system is initialized so that a fast streaming can start when a request arrives from the client terminal, and

said template file is comprised of a media file directory from which the media are taken, a media type, a logical start time and a logical end time, so that said template file is used by the synchronization engine 1426 which refers to the template file to set up a basic synchronization strategy when the client Java Applet is firstly initialized, and a user can make up the template file with a general text-based editor.

5. The dynamic multimedia web cataloging system using Java of claim 1, wherein said media to be reproduced within a respective corresponding logical time is classified into video, audio, text, 2D still image, 2D image set for 3D modeling, animation image and HTML data, which is independently performed at a mutually logical reproduction time in the window of the client Java Applet and other web browser, in which the 2D image set for 3D modeling is used in independent form for making a set of 3D modeling image and animation in the image player, and all the other media except for the 2D image set for 3D modeling are comprised of frames divided in units of an independent reproduction time.

6. The dynamic multimedia web cataloging system using Java of claim 5, wherein each of the media is reproduced independently or any one of one or more same media and/or different media combinations when said web cataloging is dynamically performed, and wherein said media combination is one of a first media combination of video and textual caption, a second media combination of video and audio, a third media combination of video, audio, and textual caption, a fourth media combination of video, audio, textual caption, and still image, a fifth media combination of video, audio, first language textual caption, and second language textual caption, a sixth media combination of video, audio, first language textual caption, second language textual caption, and third textual caption, etc., a seventh media combination of video, audio, and HTML document, an eighth media combination of video, audio, textual caption, and HTML document, a ninth media combination of video, audio, textual caption, HTML document, and still image, a tenth media combination of still image, and textual caption for realizing a slide show, an eleventh media combination of still image, textual caption, and audio per each image for realizing an audio slide show, a twelfth media combination of animation image for realizing a rolling banner

advertisement, a thirteenth media combination of animation image, and audio per each image for realizing a sound rolling banner advertisement, a fourteenth media combination of 3D panorama image, a fifteenth media combination of continuity of 2D images photographed at different angles for realizing 3D photo modeling effect and a sixteenth combination of animation image, textual caption, and audio per each image for realizing a sound/textual caption rolling banner advertisement.

7. The dynamic multimedia web cataloging system using Java of claim 1, wherein said media player engine is comprised of at least one media player produced according to a flag set with respect to the media which is selected for web catalog representation by a user after a user parameter input via the web browser is parsed, and said media player is comprised of a video player, an audio player, an image player, a text player and a HTML loader, in which these media are simultaneously produced and thus one or more media are simultaneously represented.

8. The dynamic multimedia web cataloging system using Java of claim 1, wherein said media display engine comprises a Java AWT (abstract windows toolkit) API (application program interface) which is connected to a video display device and takes charge of visual representation, a Java audio API which is connected to an audio output device and takes charge of audio output, and a web browser API 42810 taking charge of the HTML output.

9. The dynamic multimedia web cataloging system using Java of claim 1, further comprising at least one contents servers in which particular media data is dispersedly arranged considering the performance of the server system and the load of the network, and a mutual physical storage position is shared based on a media file directory and a TCP/IP.

10. A dynamic multimedia web cataloging method using Java under a client/server system circumstance where a client terminal including a web browser and a Java virtual machine and a server system including a web server are connected via the Internet network, the Java-based dynamic multimedia web cataloging method comprising the steps of:

(a) gaining access to the web server of the server system using the web browser;

(b) downloading a client Java Applet information contained HTML catalog document to the web browser from the web server;

(c) parsing a HTML tag of the HTML catalog document in the web browser, requesting to the server system multimedia data including at least one of video, audio, 2D/3D image, animation, text and HTML data necessary for a multimedia web cataloging via a Java virtual machine, creating a client Java Applet object including a synchronization engine for combining the multimedia data received via the web server dynamically, to then reproduce the combined result and a media reception engine for receiving a cataloging multimedia data packet from the server system in a streaming method and storing the received cataloging multimedia data packet in a media queue, and creating a graphic user interface (GUI) in the determined HTML document;

(d) parsing a user input parameter if a user initial input value is input via the web browser and then establishing flags with respect to media selected for a user desired catalog representation;

(e) decoding each media data received in correspondence to each media selected based on the established representation information, in accordance with a format of each file, creating a media player object including a video player, audio player, an image player, a text player and a HTML player to be reproduced in a corresponding arrangement device, and

allocating a media queue corresponding to each media player object in a public type;

(f) configuring initialization input data for starting a catalog representation in the client Java Applet in the case that the media player object has been created to be handed over to a Java event handler;

(g) transferring a velocity value, a reproduction position and a frame rate in units of frame per second which are established according to a velocity control, and a selection of a reproduction position and a reproduction operation based on the parsed input value, from the synchronization engine to the server system via the web server;

(h) parsing the contents in a video server engine upon a request of video data according to the request and streaming the parsed result to the client Java Applet via the web server and reading audio, image, text and HTML data except for video data from a template file system directly in the web server and streaming the read result to the client Java Applet;

(i) receiving a cataloging multimedia data packet in the media reception engine in a streaming method and storing the received cataloging multimedia data packet in a media queue;

(j) synchronizing each received media data under the control of a reproduction time of a synchronization engine in a media player, to be decoded in accordance with a format of each file to be reproduced in a corresponding arrangement device; and

(k) displaying the media data decoded in the media player by a display engine.

11. The dynamic multimedia web cataloging method using Java of claim 10, wherein said template file in said multimedia template file system comprises information about a storage position, a media type, a media name, a representation start time and a representation end time in the web server, with respect to each media data, and

said template file is used by the synchronization engine which refers to the template file to set up a basic synchronization strategy when the client Java Applet is firstly initialized, and

further comprising the step of loading said video template file in the memory when a video server servlet by the web server.

12. The dynamic multimedia web cataloging method using Java of claim 10, wherein said video data streaming step comprises the sub-steps of:

executing a session connection of video servlet in the client terminal by the user and transferring a command including parameters with respect to user desired video title and the video data to a video server engine;

parsing the video title from the input command, selecting a corresponding template object of the memory with respect to the parsed title, and sequentially judging whether a reproduction position, a reproduction velocity and a frame rate (FPS; frames per second) from which the video data are read from the input parameter are altered; and

reading a video stream file at a current velocity from an actual current offset after completing a parsing process with respect to all input parameters, packetizing a stream as many as the number of the current FPS as soon as the video data has been read, and transferring the packet to the client terminal.

13. The dynamic multimedia web cataloging method using Java of claim 10, wherein said catalog media data packet receiving step comprises the steps of:

requesting a packet of predetermined set time to the server system in advance in order to calculate the bandwidth of the current network, in the case that the media reception engine object has been produced, receiving the packet of the set time from the server system and then calculating an average reception time value per second;

calculating an optimal FPS at the current network state with the average reception time value;

entering an asynchronous listen state until data is transmitted from the actual server system, once the optimal FPS is calculated at the current state and judging whether a data packet arrives from the server system; and

receiving a data packet every one second once the data packet arrives in the result of judgement, parsing the media type from the header portion of the received next packet, and inserting the data packet in the corresponding media queue.

14. The dynamic multimedia web cataloging method using Java of claim 13, further comprising the steps of:

judging whether the logical end time of the received last packet exceeds the logical time, that is, the reproduction position under the current reproduction, if a bandwidth situation becomes worse suddenly during reception, when the successively received packets are inserted into the media queue;

setting the reproduction velocity of the synchronization engine to be zero and entering a block state for a moment in the case that the logical

end time of the last packet exceeds the logical time, that is, the reproduction position under the current reproduction in the result of judgement;

predicting an optimal buffering time based on the average reception value per second at the block set state, receiving a data packet as much as the buffering time, and inserting the received data packet into the media queue; and

resuming a reproduction from the stop point after the synchronization reproduction velocity is set one again if the buffering step is completed, and recalculating the average reception time per second every predetermined set time.

15. The dynamic multimedia web cataloging method using Java of claim 10, wherein in said synchronization step of the synchronization engine with respect to said media data, a logical reproduction time of each data frame of video is determined based on the FPS, that of audio is determined according to a sampling rate per second, and those of media combined using a start time, end time and reproduction velocity information of the corresponding template file in case of the discontinuous media including image, text and HTML data are determined, in order to perform synchronization.

16. The dynamic multimedia web cataloging method using Java of claim 10, wherein said media data is controlled so that the media data is reproduced by a virtual VCR function of the synchronization engine.

17. The dynamic multimedia web cataloging method using Java of claim 16, further comprising the step of opening the object reference, reproduction velocity and logical time to each media player, in such a manner that each media player can refer to a synchronization variable and flag with respect to the current synchronization situation, after the

synchronization engine transfers the reproduction velocity, reproduction position and the frame rate per second (FPS) to the server system according to the input from the user.

18. The dynamic multimedia web cataloging method using Java of claim 10, wherein said media player object produced according to the selected media are generated simultaneously and in parallel by a multithread function of Java.

19. The dynamic multimedia web cataloging method using Java of claim 10, further comprising the step of transferring the altered velocity value, reproduction position and frame rate per second to the server system, in the case that the user alters the set velocity value, the set reproduction position and the set frame rate per second.

20. The dynamic multimedia web cataloging method using Java of claim 10, wherein said video decoding step of said video player object comprises the steps of:

generating a video decoder object for converting the synchronization information into pixel images which can be output in an actual media display engine as a thread object;

comparing the reproduction velocity and the reproduction position of the synchronization engine, and the logical start time and the logical end time in the actual video frames in the generated video decoder object, to check whether there are video frames to be reproduced at a current point in time in the media queue;

if the velocity of the synchronization engine is zero, or if the velocity of the synchronization engine is not zero and there is no video frame corresponding to the current logical point in time, sleeping its thread as much as the set time and if there is the video frame corresponding to the current

logical point in time in the media queue, taking the video frame from the media queue;

performing MPEG-1 decoding with respect to the video frame taken from the media queue;

after the decoding has been completed, outputting the decoded result on a screen via a video device of the display engine; and

after a video decoder thread is slept for a remaining frame reproduction time so that the decoded output can be displayed on the screen for a reproduction time of the current frame, automatically erasing the output video frame from the media queue.

21. The dynamic multimedia web cataloging method using Java of claim 10, wherein said audio decoding step of said audio player object comprises the steps of:

generating an audio decoder object for converting the synchronization information into audio data which can be output in an actual media display engine as a thread object;

comparing the reproduction velocity and the reproduction position of the synchronization engine, and the logical start time and the logical end time in the actual audio frames in the generated audio decoder object, to check whether there are audio frames to be reproduced at a current point in time in the media queue;

if the velocity of the synchronization engine is zero, or if the velocity of the synchronization engine is not zero and there is no audio frame corresponding to the current logical point in time, sleeping its thread as much as the set time and if there is the audio frame corresponding to the current logical point in time in the media queue, taking the audio frame from the media queue;

performing GSM decoding with respect to the audio frame taken from the media queue;

after the decoding has been completed, outputting the decoded result through a speaker in an audio device of the display engine; and

after an audio decoder thread is slept for a remaining frame reproduction time so that the decoded output can be displayed on the speaker for a reproduction time of the current frame, automatically erasing the output audio frame from the media queue.

22. The dynamic multimedia web cataloging method using Java of claim 10, wherein said image decoding step in said image player object comprises the steps of:

generating an image decoder object for converting the synchronization information into pixel images which can be output by the actual medial display engine, together with sub-decoding component objects which can selectively handle a 2D image modeling and a 3D image modeling according to a media application type, as a thread object;

comparing the reproduction velocity and the reproduction position of the synchronization engine, and the logical start time and the logical end time in the actual image frames in the generated image decoder object, to check whether there are image frames to be reproduced at a current point in time in the media queue;

if the velocity of the synchronization engine is zero, or if the velocity of the synchronization engine is not zero and there is no image frame corresponding to the current logical point in time, sleeping its thread as much as the set time and if there is the image frame corresponding to the current logical point in time in the media queue, taking the image frame from the media queue;

reading the actual image data from the server system in real time and combining the read result with a logical reproduction time to thereby complete a perfect image frame, based on the information about a media file directory from which media are taken, a media type, a logical start

time and a logical end time, which are included in the image frame taken from the media queue;

after judging whether the completed media frame is 2D or 3D, performing a 2D image decoding with respect to the media frame pieces in the case of 2D and performing a 3D image modeling in the case of 3D;

after the decoding is completed, outputting the decoded result on a screen via the video device of the display engine; and

after the image decoder object thread is slept as much as the remaining frame reproduction time so that the output can be displayed on the screen for the reproduction time of the current frame, automatically erasing the output image frames for the media queue.

23. The dynamic multimedia web cataloging method using Java of claim 22, wherein said 3D image modeling comprises a motion module which receives the successive 2D images of 18 sheets or more which have been photographed at a different angle, in sequence and performs a three dimensional modeling, and panorama module which implements a three dimensional 360-degree panning effect with a sheet of 2D image in which when a particular space in the image is clicked, the panorama module includes a function of linking the image with a different media player.

24. The dynamic multimedia web cataloging method using Java of claim 10, wherein each of the media is reproduced independently or any one of one or more same media and/or different media combinations when said web cataloging is dynamically performed, and wherein said media combination is one of a first media combination of video and textual caption, a second media combination of video and audio, a third media combination of video, audio, and textual caption, a fourth media combination of video, audio, textual caption, and still image, a fifth media combination of video, audio, first language textual caption, and second language textual caption, a

sixth media combination of video, audio, first language textual caption, second language textual caption, and third textual caption, etc., a seventh media combination of video, audio, and HTML document, an eighth media combination of video, audio, textual caption, and HTML document, a ninth media combination of video, audio, textual caption, HTML document, and still image, a tenth media combination of still image, and textual caption for realizing a slide show, an eleventh media combination of still image, textual caption, and audio per each image for realizing an audio slide show, a twelfth media combination of animation image for realizing a rolling banner advertisement, a thirteenth media combination of animation image, and audio per each image for realizing a sound rolling banner advertisement, a fourteenth media combination of 3D panorama image, a fifteenth media combination of continuity of 2D images photographed at different angles for realizing 3D photo modeling effect and a sixteenth combination of animation image, textual caption, and audio per each image for realizing a sound/textual caption rolling banner advertisement.

25. The dynamic multimedia web cataloging method using Java of claim 24, wherein said video is MPEG-1 data and said audio is real-time GSM data.

26. The dynamic multimedia web cataloging method using Java of claim 10, wherein said client terminal is one of a general portable phone, a personal digital assistant (PDA), a settop box, a digital TV or a web phone which is porting a personal Java (PersonalJava) software, and a Java operating system (JavaOs) mounted Java chip, as well as a personal computer (PC), a workstation, a notebook PC or a palm top PC which is designed based on a pure-Java and is mounted with a Java virtual machine and a web browser.

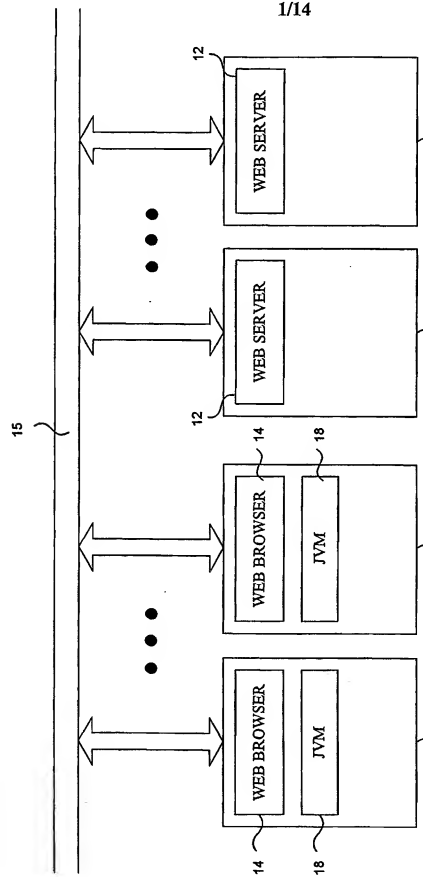

15
12
WEB SERVER
SSn
12
WEB SERVER
SS1
14
18
WEB BROWSER
JVM
Cn
14
18
WEB BROWSER
JVM
C1

Fig. 1

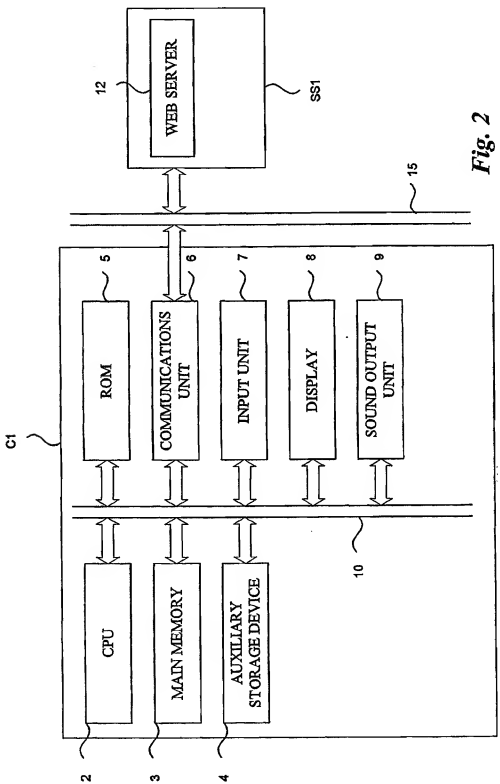

WEB SERVER
12
SS1
15
C1
ROM
5
COMMUNICATIONS UNIT
6
INPUT UNIT
7
DISPLAY
8
SOUND OUTPUT UNIT
9
10
CPU
MAIN MEMORY
AUXILIARY STORAGE DEVICE
2
3
4

Fig. 2

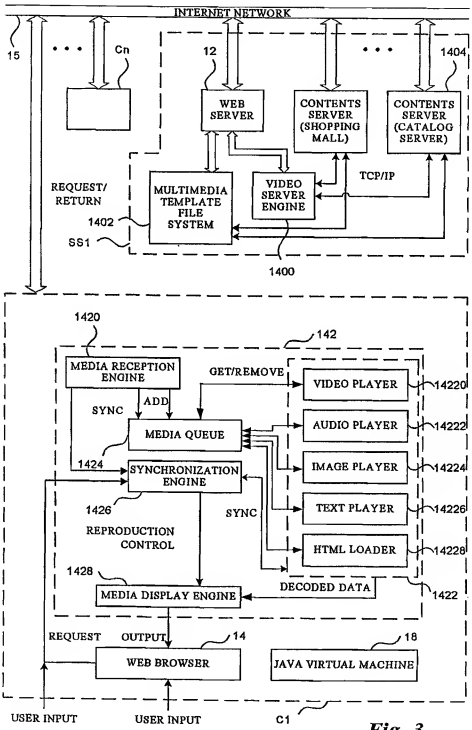

INTERNET NETWORK
15
Cn
12
1404
WEB SERVER
CONTENTS SERVER (SHOPPING MALL)
CONTENTS SERVER (CATALOG SERVER)
REQUEST/ RETURN
MULTIMEDIA TEMPLATE FILE SYSTEM
VIDEO SERVER ENGINE
TCP/IP
1402
SS1
1400
1420
142
MEDIA RECEPTION ENGINE
GET/REMOVE
VIDEO PLAYER
14220
ADD
SYNC
MEDIA QUEUE
AUDIO PLAYER
14222
1424
SYNCHRONIZATION ENGINE
IMAGE PLAYER
14224
1426
SYNC
TEXT PLAYER
14226
REPRODUCTION CONTROL
HTML LOADER
14228
1428
MEDIA DISPLAY ENGINE
DECODED DATA
1422
REQUEST
OUTPUT
14
18
WEB BROWSER
JAVA VIRTUAL MACHINE
USER INPUT
USER INPUT
C1

Fig. 3

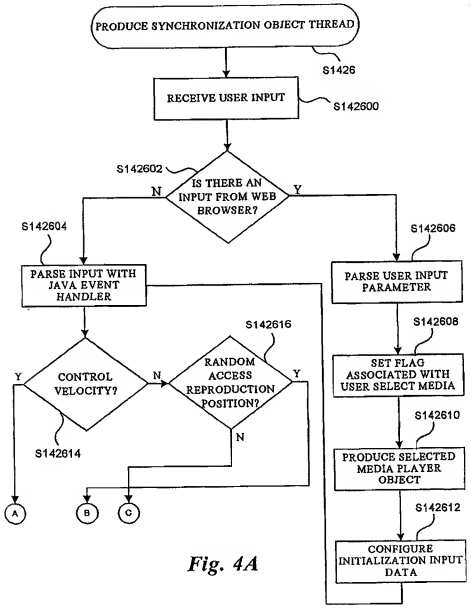

PRODUCE SYNCHRONIZATION OBJECT THREAD
S1426
RECEIVE USER INPUT
S142600
S142602
IS THERE AN INPUT FROM WEB BROWSER?
N
Y
S142604
PARSE INPUT WITH JAVA EVENT HANDLER
S142606
PARSE USER INPUT PARAMETER
S142608
SET FLAG ASSOCIATED WITH USER SELECT MEDIA
S142610
PRODUCE SELECTED MEDIA PLAYER OBJECT
S142612
CONFIGURE INITIALIZATION INPUT DATA
S142616
RANDOM ACCESS REPRODUCTION POSITION?
CONTROL VELOCITY?
S142614
A
B
C

Fig. 4A

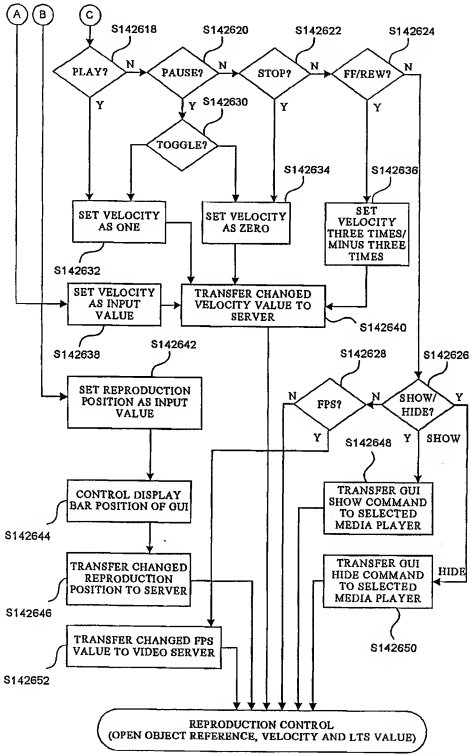

A
B
C
S142618
PLAY?
N
Y
S142620
PAUSE?
N
Y
S142622
STOP?
N
Y
S142624
FF/REW?
N
Y
S142630
TOGGLE?
Y
S142634
S142636
SET VELOCITY AS ONE
SET VELOCITY AS ZERO
SET VELOCITY THREE TIMES/ MINUS THREE TIMES
S142632
SET VELOCITY AS INPUT VALUE
TRANSFER CHANGED VELOCITY VALUE TO SERVER
S142640
S142638
S142642
S142628
S142626
SET REPRODUCTION POSITION AS INPUT VALUE
FPS?
N
Y
SHOW/ HIDE?
N
Y
SHOW
S142648
CONTROL DISPLAY BAR POSITION OF GUI
TRANSFER GUI SHOW COMMAND TO SELECTED MEDIA PLAYER
S142644
TRANSFER CHANGED REPRODUCTION POSITION TO SERVER
TRANSFER GUI HIDE COMMAND TO SELECTED MEDIA PLAYER
HIDE
S142646
S142650
TRANSFER CHANGED FPS VALUE TO VIDEO SERVER
S142652
REPRODUCTION CONTROL
(OPEN OBJECT REFERENCE, VELOCITY AND LTS VALUE)

Fig. 4B

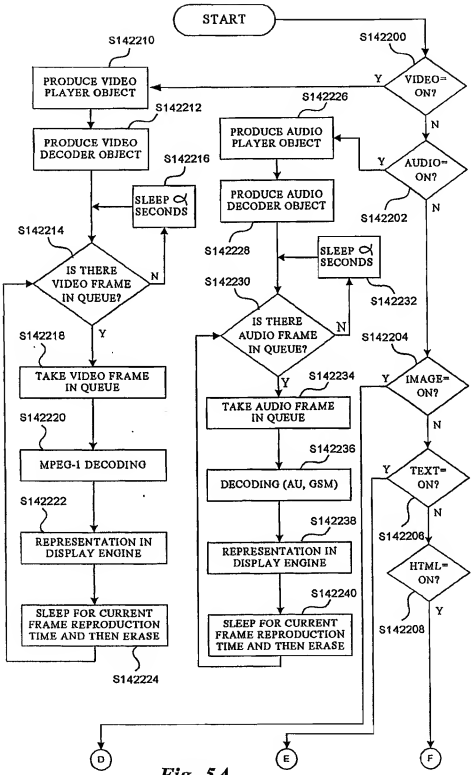

START
S142200
VIDEO= ON?
Y
N
S142210
PRODUCE VIDEO PLAYER OBJECT
S142212
PRODUCE VIDEO DECODER OBJECT
S142216
SLEEP X SECONDS
S142214
IS THERE VIDEO FRAME IN QUEUE?
N
Y
S142218
TAKE VIDEO FRAME IN QUEUE
S142220
MPEG-1 DECODING
S142222
REPRESENTATION IN DISPLAY ENGINE
SLEEP FOR CURRENT FRAME REPRODUCTION TIME AND THEN ERASE
S142224
S142226
PRODUCE AUDIO PLAYER OBJECT
AUDIO= ON?
Y
N
S142202
PRODUCE AUDIO DECODER OBJECT
S142228
SLEEP X SECONDS
S142232
S142230
IS THERE AUDIO FRAME IN QUEUE?
N
Y
S142234
TAKE AUDIO FRAME IN QUEUE
S142236
DECODING (AU, GSM)
S142238
REPRESENTATION IN DISPLAY ENGINE
S142240
SLEEP FOR CURRENT FRAME REPRODUCTION TIME AND THEN ERASE
S142204
IMAGE= ON?
Y
N
TEXT= ON?
Y
N
S142206
HTML= ON?
Y
S142208
D
E
F

Fig. 5A

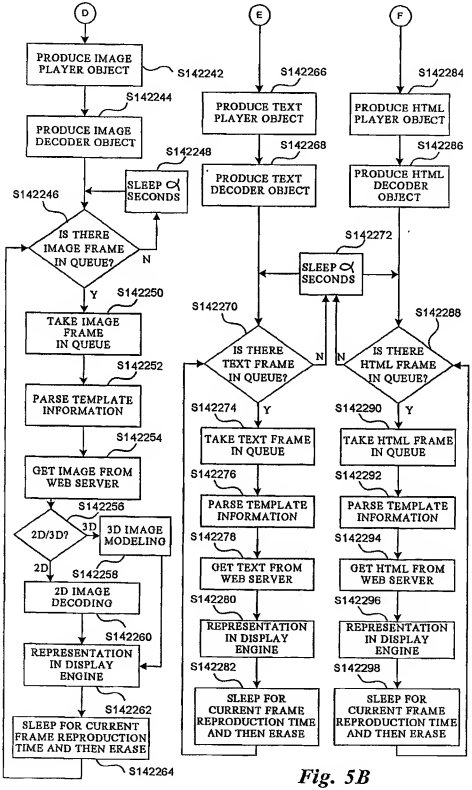

D
E
F
PRODUCE IMAGE PLAYER OBJECT
S142242
PRODUCE IMAGE DECODER OBJECT
S142244
SLEEP X SECONDS
S142248
S142246
IS THERE IMAGE FRAME IN QUEUE?
N
Y
TAKE IMAGE FRAME IN QUEUE
S142250
PARSE TEMPLATE INFORMATION
S142252
GET IMAGE FROM WEB SERVER
S142254
2D/3D?
3D
3D IMAGE MODELING
S142256
2D
2D IMAGE DECODING
S142258
REPRESENTATION IN DISPLAY ENGINE
S142260
SLEEP FOR CURRENT FRAME REPRODUCTION TIME AND THEN ERASE
S142262
S142264
PRODUCE TEXT PLAYER OBJECT
S142266
PRODUCE TEXT DECODER OBJECT
S142268
IS THERE TEXT FRAME IN QUEUE?
S142270
N
Y
TAKE TEXT FRAME IN QUEUE
S142274
PARSE TEMPLATE INFORMATION
S142276
GET TEXT FROM WEB SERVER
S142278
REPRESENTATION IN DISPLAY ENGINE
S142280
SLEEP FOR CURRENT FRAME REPRODUCTION TIME AND THEN ERASE
S142282
PRODUCE HTML PLAYER OBJECT
S142284
PRODUCE HTML DECODER OBJECT
S142286
SLEEP X SECONDS
S142272
IS THERE HTML FRAME IN QUEUE?
S142288
N
Y
TAKE HTML FRAME IN QUEUE
S142290
PARSE TEMPLATE INFORMATION
S142292
GET HTML FROM WEB SERVER
S142294
REPRESENTATION IN DISPLAY ENGINE
S142296
SLEEP FOR CURRENT FRAME REPRODUCTION TIME AND THEN ERASE
S142298

Fig. 5B

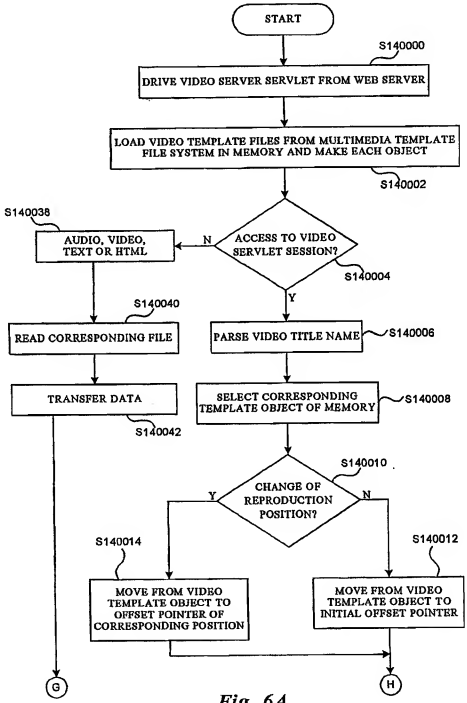

START
S140000
DRIVE VIDEO SERVER SERVLET FROM WEB SERVER
LOAD VIDEO TEMPLATE FILES FROM MULTIMEDIA TEMPLATE FILE SYSTEM IN MEMORY AND MAKE EACH OBJECT
S140002
S140038
AUDIO, VIDEO, TEXT OR HTML
N
ACCESS TO VIDEO SERVLET SESSION?
S140004
Y
S140040
READ CORRESPONDING FILE
PARSE VIDEO TITLE NAME
S140006
TRANSFER DATA
SELECT CORRESPONDING TEMPLATE OBJECT OF MEMORY
S140008
S140042
S140010
Y
CHANGE OF REPRODUCTION POSITION?
N
S140014
S140012
MOVE FROM VIDEO TEMPLATE OBJECT TO OFFSET POINTER OF CORRESPONDING POSITION
MOVE FROM VIDEO TEMPLATE OBJECT TO INITIAL OFFSET POINTER
G
H

Fig. 6A

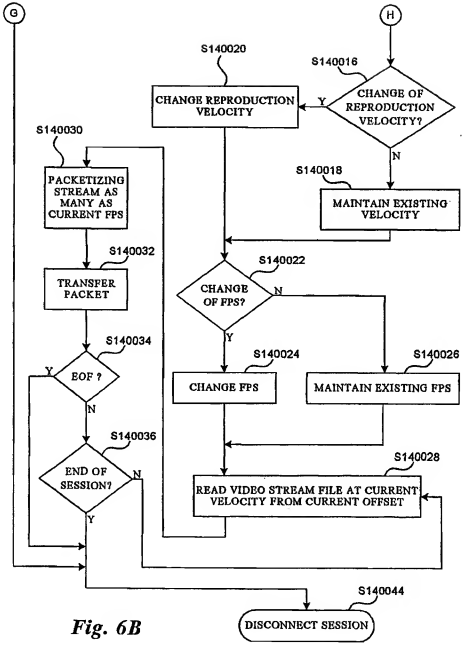

G
H
S140020
S140016
CHANGE REPRODUCTION VELOCITY
CHANGE OF REPRODUCTION VELOCITY?
Y
N
S140030
S140018
PACKETIZING STREAM AS MANY AS CURRENT FPS
MAINTAIN EXISTING VELOCITY
S140032
S140022
TRANSFER PACKET
CHANGE OF FPS?
N
Y
S140034
S140024
S140026
Y
EOF ?
CHANGE FPS
MAINTAIN EXISTING FPS
N
S140036
S140028
END OF SESSION?
N
READ VIDEO STREAM FILE AT CURRENT VELOCITY FROM CURRENT OFFSET
Y
S140044
DISCONNECT SESSION

Fig. 6B

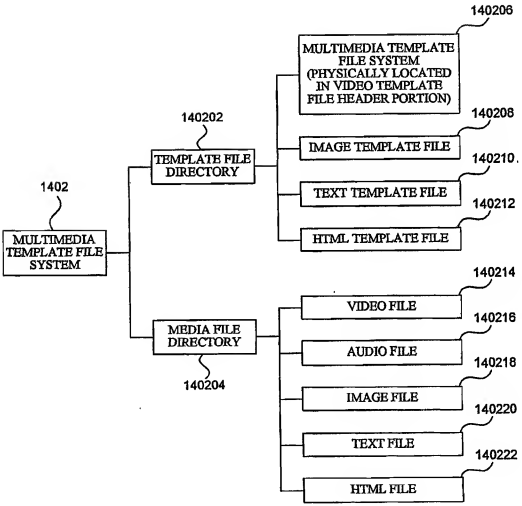

1402
MULTIMEDIA TEMPLATE FILE SYSTEM
140202
TEMPLATE FILE DIRECTORY
140204
MEDIA FILE DIRECTORY
140206
MULTIMEDIA TEMPLATE FILE SYSTEM (PHYSICALLY LOCATED IN VIDEO TEMPLATE FILE HEADER PORTION)
140208
IMAGE TEMPLATE FILE
140210
TEXT TEMPLATE FILE
140212
HTML TEMPLATE FILE
140214
VIDEO FILE
140216
AUDIO FILE
140218
IMAGE FILE
140220
TEXT FILE
140222
HTML FILE

Fig. 7

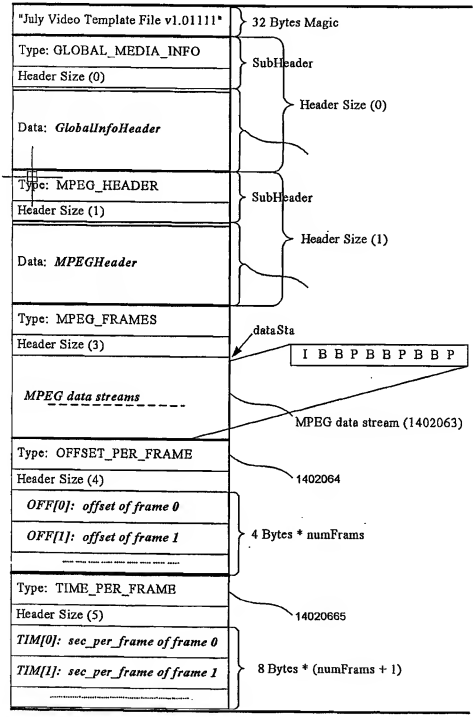

"July Video Template File v1.01111"
32 Bytes Magic
Type: GLOBAL_MEDIA_INFO
Header Size (0)
SubHeader
Data: GlobalInfoHeader
Header Size (0)
Type: MPEG_HEADER
Header Size (1)
SubHeader
Data: MPEGHeader
Header Size (1)
Type: MPEG_FRAMES
Header Size (3)
dataSta
I B B P B B P B B P
MPEG data streams
MPEG data stream (1402063)
Type: OFFSET_PER_FRAME
Header Size (4)
1402064
OFF[0]: offset of frame 0
OFF[1]: offset of frame 1
4 Bytes * numFrams
Type: TIME_PER_FRAME
Header Size (5)
14020665
TIM[0]: sec_per_frame of frame 0
TIM[1]: sec_per_frame of frame 1
8 Bytes * (numFrams + 1)

Fig. 8

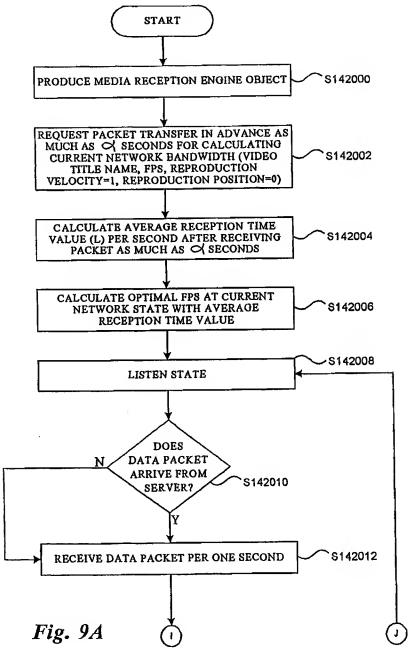

START
PRODUCE MEDIA RECEPTION ENGINE OBJECT
S142000
REQUEST PACKET TRANSFER IN ADVANCE AS MUCH AS α SECONDS FOR CALCULATING CURRENT NETWORK BANDWIDTH (VIDEO TITLE NAME, FPS, REPRODUCTION VELOCITY=1, REPRODUCTION POSITION=0)
S142002
CALCULATE AVERAGE RECEPTION TIME VALUE (L) PER SECOND AFTER RECEIVING PACKET AS MUCH AS α SECONDS
S142004
CALCULATE OPTIMAL FPS AT CURRENT NETWORK STATE WITH AVERAGE RECEPTION TIME VALUE
S142006
LISTEN STATE
S142008
DOES DATA PACKET ARRIVE FROM SERVER?
S142010
N
Y
RECEIVE DATA PACKET PER ONE SECOND
S142012
I
J

Fig. 9A

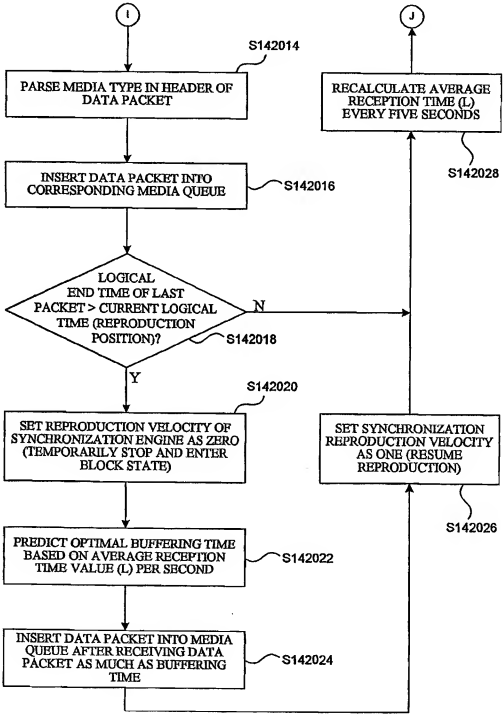

I
J
S142014
PARSE MEDIA TYPE IN HEADER OF DATA PACKET
INSERT DATA PACKET INTO CORRESPONDING MEDIA QUEUE
S142016
LOGICAL END TIME OF LAST PACKET > CURRENT LOGICAL TIME (REPRODUCTION POSITION)?
S142018
N
Y
S142020
SET REPRODUCTION VELOCITY OF SYNCHRONIZATION ENGINE AS ZERO (TEMPORARILY STOP AND ENTER BLOCK STATE)
PREDICT OPTIMAL BUFFERING TIME BASED ON AVERAGE RECEPTION TIME VALUE (L) PER SECOND
S142022
INSERT DATA PACKET INTO MEDIA QUEUE AFTER RECEIVING DATA PACKET AS MUCH AS BUFFERING TIME
S142024
SET SYNCHRONIZATION REPRODUCTION VELOCITY AS ONE (RESUME REPRODUCTION)
S142026
RECALCULATE AVERAGE RECEPTION TIME (L) EVERY FIVE SECONDS
S142028

Fig. 9B

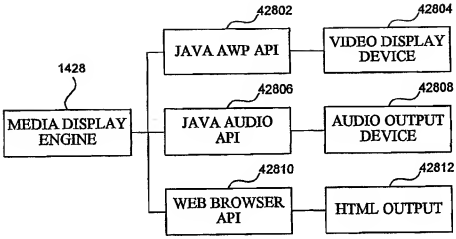

42802
42804
JAVA AWP API
VIDEO DISPLAY DEVICE
1428
42806
42808
MEDIA DISPLAY ENGINE
JAVA AUDIO API
AUDIO OUTPUT DEVICE
42810
42812
WEB BROWSER API
HTML OUTPUT

Fig. 10

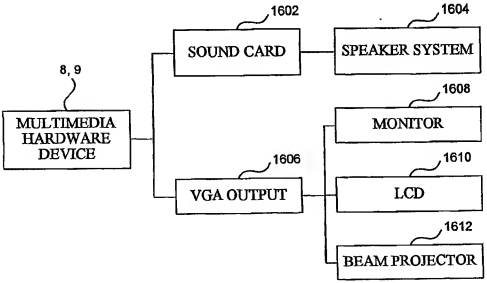

1602
1604
SOUND CARD
SPEAKER SYSTEM
8, 9
1608
MULTIMEDIA HARDWARE DEVICE
MONITOR
1606
1610
VGA OUTPUT
LCD
1612
BEAM PROJECTOR

Fig. 11

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/77897 A3

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US00/26508

(22) International Filing Date:
26 September 2000 (26.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10-1999-0041359
27 September 1999 (27.09.1999) KR

(71) Applicant: TORVEO TECHNOLOGIES [US/US]; 201 Marshall Street, Redwood City, CA 94063 (US).

(72) Inventors: HEO, Hong; 523-10, Sangmunl-dong, Dobong-gu, Seoul (KR). PARK, Sun, Eun; 101 dong 608ho, Kolong APT, Bangy-dong, Songpa-gu, Seoul (KR). LEE, Jin, Bae; 502ho, 547-3 Dogok-dong, Kangnam-gu, Seoul (KR). KANG, Hee, Kook; 201ho, 463-43, Chunho2-dong, Kangdong-gu, Seoul (KR). SEO, Seong, Cheol; 211 dong 101ho, Donga APT, Keoye-dong, Songpa-gu, Seoul (KR).

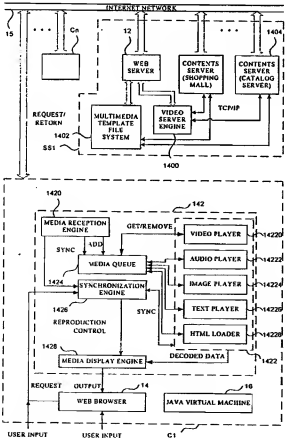
(74) Agents: PIRIO, Maurice, J. et al.; Perkins Coie LLP, P.O. Box 1247, Seattle, WA 98111-1247 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

[Continued on next page]

(54) Title: SYSTEM AND METHOD USING A WEB CATALOG WITH DYNAMIC MULTIMEDIA DATA USING JAVA



(57) Abstract: A Java-based dynamic multimedia web cataloging system for expressing catalog contents most effectively with a web browser in which various multimedia data having streamed from a server system to a web browser is mixed with HTML documents. The system includes a web server for providing a client Java Applet information contained HTML catalog document to the web browser when user accesses the server system via the web browser, a video server engine for streaming the video data on a real-time basis, a template file system including template files having overall information appropriate for a streaming action, and the client Java Applet which is inserted in the HTML catalog document when the client terminal accesses the web server, downloaded in the web browser, and created by the Java virtual machine, for implementing functions necessary for a web cataloging dynamically on a real-time basis. The client Java Applet includes a synchronization engine, a media reception engine, a media player engine, and a display engine.

WO 01/77897 A3

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
2 May 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

I' ERNATIONAL SEARCH REPORT

In..ational Application No
PCT/US 00/26508

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

INSPEC, EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	ENGLAND P ET AL: "RAVE: real-time services for the Web" FIFTH INTERNATIONAL WORLD WIDE WEB CONFERENCE, PARIS, FRANCE, 6-10 MAY 1996, vol. 28, no. 7-11, pages 1547-1558, XP004018250 Computer Networks and ISDN Systems, May 1996, Elsevier, Netherlands ISSN: 0169-7552 abstract page 1547, left-hand column, line 12 -page 1548, right-hand column, line 5 page 1549, right-hand column, line 1 -page 1551, left-hand column, line 31 page 1552, left-hand column, line 9 -page 1553, right-hand column, line 4 page 1556, left-hand column, line 6 -page 1556, right-hand column, line 17 --- -/--	1-6, 8-13, 16-18, 24-26

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 October 2001	19/11/2001

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer
Boyadzhiev, Y

I ERNATIONAL SEARCH REPORT

Inte..ational Application No
PCT/US 00/26508

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 98 59460 A (RAZ URI) 30 December 1998 (1998-12-30) page 2, line 15 -page 3, line 23 page 8, line 11 -page 12, line 9 page 14, line 20 -page 19, line 14 page 20, line 25 -page 21, line 30 page 23, line 12 -page 24, line 16 ---	1-6, 8-13, 16-18, 24-26
A	HERNG-YOW CHEN ET AL: "Design of a Web-based synchronized multimedia lecture system for distance education" PROCEEDINGS IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS, PROCEEDINGS OF ICMCS99: IEEE MULTIMEDIA SYSTEMS '99: INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS, FLORENCE, ITALY, 7-11 JUNE 1999, pages 887-891 vol.2, XP002181582 1999, Los Alamitos, CA, USA, IEEE Comput. Soc, USA ISBN: 0-7695-0253-9 abstract page 888, left-hand column, line 1 -page 888, right-hand column, line 4 page 889, left-hand column, line 33 -page 890, right-hand column, line 43 ---	1-26
A	CHEN ET AL: "Real Time Video and Audio in the World Wide Web" INTERNATIONAL WORLD WIDE WEB CONFERENCE, XX, XX, 11 December 1995 (1995-12-11), pages 1-14, XP002107740 abstract page 1, line 1 -page 3, line 44 page 6, line 12 -page 7, line 14 page 8, line 24 -page 9, line 24 page 11, line 27 -page 12, line 16 ---	1-26
A	KRISTENSEN A: "Template resolution in XML/HTML" COMPUTER NETWORKS AND ISDN SYSTEMS, NORTH HOLLAND PUBLISHING. AMSTERDAM, NL, vol. 30, no. 1-7, 1 April 1998 (1998-04-01), pages 239-249, XP004121423 ISSN: 0169-7552 abstract page 239, left-hand column, line 1 -page 241, right-hand column, line 19 page 245, left-hand column, line 40 -page 246, left-hand column, line 15 -----	1-6, 9-12,18, 21-26

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/26508

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9859460	A	30-12-1998	US	6292827 B1	18-09-2001
			EP	0990327 A1	05-04-2000
			WO	9859460 A1	30-12-1998